

--	--	--	--	--	--	--	--	--	--

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2017/2018

ECP2036 – MICROPROCESSOR SYSTEMS AND INTERFACING (ME)

21 OCTOBER 2017
9:00 AM – 11:00 AM
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 8 pages with 5 questions only.
2. Attempt **ALL** questions. All questions carry equal marks and the distribution of the marks for each question is given.
3. Please write all your answers in the Answer Booklet provided.
4. Opcode map and Special Function Register formats are provided in Appendices.

Question 1

- (a) State TWO(2) I/O ports that are unavailable for general-purpose I/O when an external memory is connected. Explain why. [2 Marks]

- (b) Answer the following questions:

- (i) What is the active register bank after execution of the following instruction?
Find out the address of R6 of that register bank.

MOV PSW, #0C9H [3 marks]

- (ii) What bit addresses are set to 1 as a result of the following instruction?

MOV R0, #26H
MOV @R0, #7AH [3 marks]

- (iii) Name the 8051 control bus signals that need to be activated while interface with external EPROM [2 marks]

- (c) Design an 8051 microcontroller based system which requires a total of 1 Kbytes of data memory. Memory blocks are available as follows:

- 256 bytes RAM
- 512 bytes ROM

Draw the configuration of the systems showing the 8051 signal lines to be used for address, data and control buses. Write the starting and ending address from each memory block. (*Hint: Use decoder where necessary*) [10 marks]

Question 2

- (a) Name TWO (2) instruction types and give an example for each instruction type listed. [2 marks]

- (b) Examine the instruction sequence below:

Line	PC	Instructions	
01	0100	ORG	0100H
02	0100	MOV	A, #45H
03	0102	MOV	B, #0A1H
04	0105	ADD	A, B
05	0107	HERE: MOV	R0, #40H
06	0109	XCHD	A, @R0
07	010A	MOV	@R0, B
08	010C	XRL	A, #50H
09	010D	SWAP	A
10	010E	RLC	A
11	010F	ADD	A, @R0
12	0110	INC	A
13	0112	SJMP	HERE
14	0114	END	

Continued...

- (i) Convert the instructions at lines 2, 3, 7 and 12 into hexadecimal machine language bytes. [4 marks]
 - (ii) Calculate the relative offset for the instruction SJMP HERE and state the hexadecimal machine language bytes. [3 marks]
 - (iii) Calculate the total execution time to execute the instructions from line 2 to line 12. Assume that the crystal frequency is 16MHz. [5 marks]
 - (iv) What is the content of the accumulator (A) and register B after the completion of instruction at line 2? [2 marks]
- (c) Suppose the following memory locations contain the initial values as follows:

Memory Location	Contents
30H	55H
78H	00H
7FH	FFH

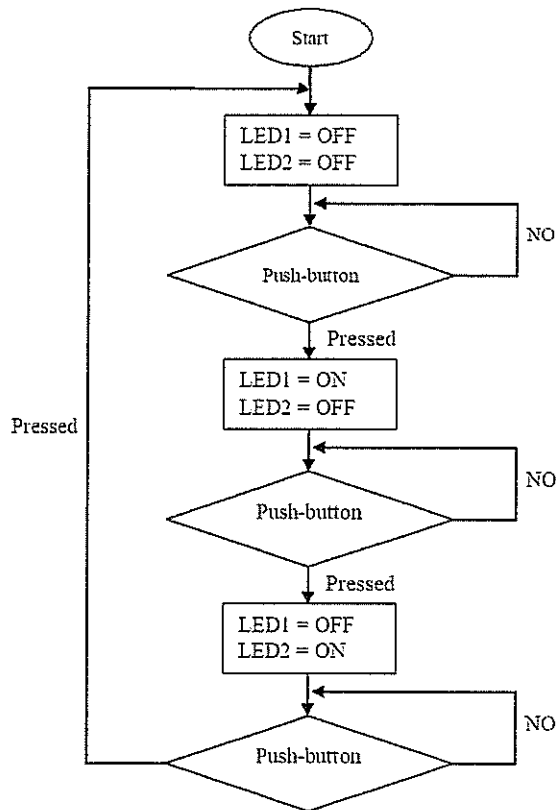
What are the contents of these memory locations after the execution of each instruction below? Why?

- (i) CPL 7FH [2 marks]
- (ii) MOV 7FH, 78H [2 marks]

Question 3

- (a) Write an 8051 programme to generate a continuous 833.33Hz square wave signal on P1.0 when P2.0 is set; otherwise if P2.1 is set, generate a continuous 1250Hz square wave signal on P1.0 instead. [12 marks]
- (b) Sketch the schematic diagram of a 8051 microcontroller system with a push-button switch connected to P0.0 and two LEDs (LED1 and LED2) connected to P1.0, and P1.1 respectively. Write an 8051 programme to turn on/off LED1, and LED2 as per described in the following flow chart.

Continued...



[8 marks]

Question 4

- (a) What are interrupt vectors? What is the first memory address available for main program when all interrupts are used? [6 marks]
- (b) Write instructions to:
- Enable the Serial interrupt, Timer 1 interrupt and External Interrupt 0. [2 marks]
 - Set Timer 1 to have higher interrupt priority [2 marks]
- (c) Write an 8051 programme to create a 10 kHz square wave on P1.0, using interrupts. [10 marks]

Question 5

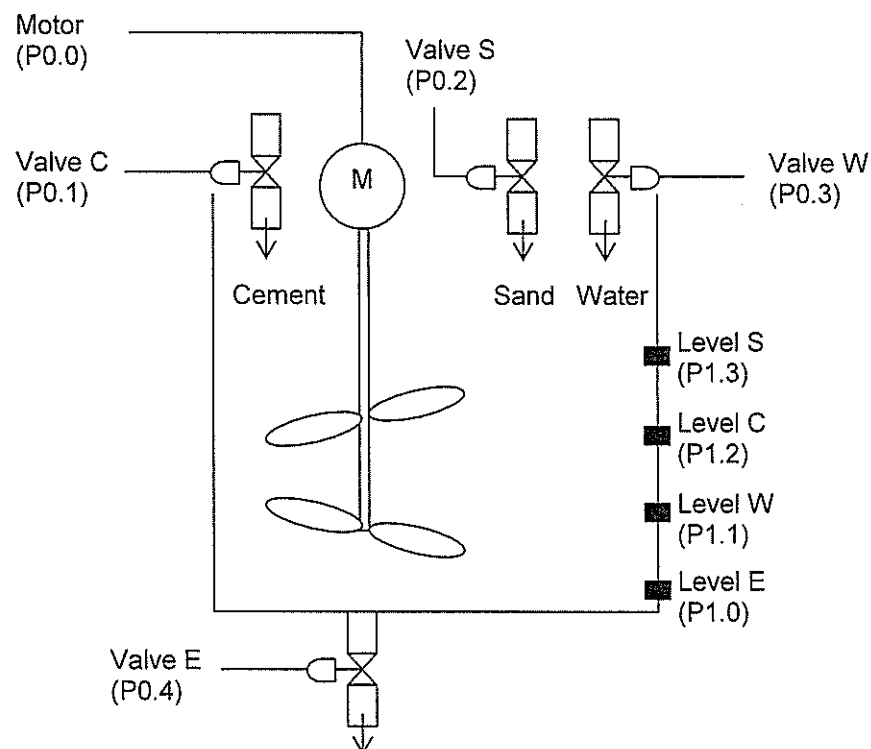
The concrete mixing system shown in the following diagram is to be controlled by an 8051 microcontroller, which involves performing the following process:

- The tank is first filled with water through a solenoid Valve W.
- When the water reaches Level W, Valve W is closed and the tank is now filled with cement through Valve C.

Continued...

- When the mixture in the tank reaches Level C, Valve C is closed and the tank is then filled with sand through Valve S.
- When the mixture in the tank reaches Level S, Valve S is closed
- The mixer motor starts for approximately 3 minutes.
- After that, the drainage Valve E opens to empty the tank.
- When the mixture reaches Level E, Valve E is closed and the whole process is repeated after 1 minute.

The tank has four level sensors that send signals to input lines P1.0 to P1.3. A logical low from the sensor indicates that the level has been reached. The output lines P0.0 to P0.3 provide signals to the solenoid valves. A logical low from the lines will open the corresponding valve. The output lines P0.4 provide signals to the mixer motor which is also activated by a logical low. Write an assembly language instruction sequence to carry out the process.



[20 marks]

Continued...

Appendix A: Opcode Map

Hex Byte	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1B, 1C NOP	3B, 2C JBC bit, rel	3B, 2C JB bit, rel	3B, 2C JNB bit, rel	2B, 2C JC rel	2B, 2C JNC rel	2B, 2C JZ rel	2B, 2C JNZ rel	2B, 2C SIMP rel	3B, 2C MOV DPTR, #16	2B, 2C ORL C, bit	2B, 2C ANL C, bit	2B, 2C PUSH dir	2B, 2C POP dir	1B, 2C MOVX A, @DPTR, A	1B, 2C MOVX @DPTR, A
1	2B, 2C AJMP (R0)	2B, 2C ACALL (R0)	2B, 2C AJMP (R1)	2B, 2C ACALL (R1)	2B, 2C AJMP (R2)	2B, 2C ACALL (R2)	2B, 2C AJMP (R3)	2B, 2C ACALL (R3)	2B, 2C AJMP (R4)	2B, 2C ACALL (R4)	2B, 2C AJMP (R5)	2B, 2C ACALL (R5)	2B, 2C AJMP (R6)	2B, 2C ACALL (R6)	2B, 2C AJMP (R7)	2B, 2C ACALL (R7)
2	3B, 2C LJMP addr16	3B, 2C LCALL addr16	1B, 1C RET	1B, 1C RETH	2B, 1C ORL dir, A	2B, 1C ANL dir, A	2B, 1C XRL dir, A	2B, 1C ORL C, bit	2B, 1C ANL C, bit	2B, 1C MOV bit, C	2B, 1C MOV A, bit	2B, 1C CPL bit	2B, 1C CLR bit	2B, 1C SETB bit	1B, 2C MOVX A, @R0	1B, 2C MOVX @R0, A
3	1B, 1C RR A	1B, 1C RRC A	1B, 1C RL A	1B, 1C RLC A	2B, 1C ORL dir, #data	2B, 1C ANL dir, #data	2B, 1C XRL dir, #data	2B, 1C ORL A, #data	2B, 1C MOV A, #data	2B, 1C MOV A, #data	2B, 1C INC DPTR	2B, 1C CPL C	2B, 1C CLR C	2B, 1C SETB C	1B, 2C MOVX A, @R1	1B, 2C MOVX @R1, A
4	1B, 1C INC A	1B, 1C DEC A	2B, 1C ADD A, #data	2B, 1C ADDC A, #data	2B, 1C ORL A, #data	2B, 1C ANL A, #data	2B, 1C XRL A, #data	2B, 1C ORL dir, #data	2B, 1C MOV dir, #data	2B, 1C MOV dir, #data	2B, 1C MUL AB	2B, 1C CJNE A, #data, rel	2B, 1C SWAP A	1B, 1C DA A	1B, 1C CLR A	1B, 1C CPL A
5	2B, 1C INC dir	2B, 1C DEC dir	2B, 1C ADD A, dir	2B, 1C ADDC A, dir	2B, 1C ORL A, dir	2B, 1C ANL A, dir	2B, 1C XRL A, dir	2B, 1C ORL dir, #data	2B, 1C MOV dir, #data	2B, 1C MOV dir, #data	2B, 1C SUBB A, dir	2B, 1C CJNE A, dir, rel	2B, 1C XCH A, dir	2B, 1C DJNZ dir, rel	2B, 1C MOV A, dir	2B, 1C MOV dir, A
6	1B, 1C INC @R0	1B, 1C DEC @R0	2B, 1C ADD A, @R0	2B, 1C ADDC A, @R0	2B, 1C ORL A, @R0	2B, 1C ANL A, @R0	2B, 1C XRL A, @R0	2B, 1C ORL @R0, #data	2B, 1C MOV @R0, #data	2B, 1C MOV @R0, #data	2B, 1C SUBB @R0, dir	2B, 1C CJNE @R0, #data, rel	2B, 1C XCH A, @R0	2B, 1C XCHD A, @R0	2B, 1C MOV A, @R0	2B, 1C MOV @R0, A
7	1B, 1C INC @R1	1B, 1C DEC @R1	2B, 1C ADD A, @R1	2B, 1C ADDC A, @R1	2B, 1C ORL A, @R1	2B, 1C ANL A, @R1	2B, 1C XRL A, @R1	2B, 1C ORL @R1, #data	2B, 1C MOV @R1, #data	2B, 1C MOV @R1, #data	2B, 1C SUBB @R1, dir	2B, 1C CJNE @R1, #data, rel	2B, 1C XCH A, @R1	2B, 1C XCHD A, @R1	2B, 1C MOV A, @R1	2B, 1C MOV @R1, A
8	1B, 1C INC R0	1B, 1C DEC R0	2B, 1C ADD A, R0	2B, 1C ADDC A, R0	2B, 1C ORL A, R0	2B, 1C ANL A, R0	2B, 1C XRL A, R0	2B, 1C ORL R0, #data	2B, 1C MOV R0, #data	2B, 1C MOV R0, #data	2B, 1C SUBB R0, dir	2B, 1C CJNE R0, #data, rel	2B, 1C XCH A, R0	2B, 1C DJNZ R0, rel	2B, 1C MOV R0, A	2B, 1C MOV A, R0
9	1B, 1C INC R1	1B, 1C DEC R1	2B, 1C ADD A, R1	2B, 1C ADDC A, R1	2B, 1C ORL A, R1	2B, 1C ANL A, R1	2B, 1C XRL A, R1	2B, 1C ORL R1, #data	2B, 1C MOV R1, #data	2B, 1C MOV R1, #data	2B, 1C SUBB R1, dir	2B, 1C CJNE R1, #data, rel	2B, 1C XCH A, R1	2B, 1C DJNZ R1, rel	2B, 1C MOV R1, A	2B, 1C MOV A, R1
A	1B, 1C INC R2	1B, 1C DEC R2	2B, 1C ADD A, R2	2B, 1C ADDC A, R2	2B, 1C ORL A, R2	2B, 1C ANL A, R2	2B, 1C XRL A, R2	2B, 1C ORL R2, #data	2B, 1C MOV R2, #data	2B, 1C MOV R2, #data	2B, 1C SUBB R2, dir	2B, 1C CJNE R2, #data, rel	2B, 1C XCH A, R2	2B, 1C DJNZ R2, rel	2B, 1C MOV R2, A	2B, 1C MOV A, R2
B	1B, 1C INC R3	1B, 1C DEC R3	2B, 1C ADD A, R3	2B, 1C ADDC A, R3	2B, 1C ORL A, R3	2B, 1C ANL A, R3	2B, 1C XRL A, R3	2B, 1C ORL R3, #data	2B, 1C MOV R3, #data	2B, 1C MOV R3, #data	2B, 1C SUBB R3, dir	2B, 1C CJNE R3, #data, rel	2B, 1C XCH A, R3	2B, 1C DJNZ R3, rel	2B, 1C MOV R3, A	2B, 1C MOV A, R3
C	1B, 1C INC R4	1B, 1C DEC R4	2B, 1C ADD A, R4	2B, 1C ADDC A, R4	2B, 1C ORL A, R4	2B, 1C ANL A, R4	2B, 1C XRL A, R4	2B, 1C ORL R4, #data	2B, 1C MOV R4, #data	2B, 1C MOV R4, #data	2B, 1C SUBB R4, dir	2B, 1C CJNE R4, #data, rel	2B, 1C XCH A, R4	2B, 1C DJNZ R4, rel	2B, 1C MOV R4, A	2B, 1C MOV A, R4
D	1B, 1C INC R5	1B, 1C DEC R5	2B, 1C ADD A, R5	2B, 1C ADDC A, R5	2B, 1C ORL A, R5	2B, 1C ANL A, R5	2B, 1C XRL A, R5	2B, 1C ORL R5, #data	2B, 1C MOV R5, #data	2B, 1C MOV R5, #data	2B, 1C SUBB R5, dir	2B, 1C CJNE R5, #data, rel	2B, 1C XCH A, R5	2B, 1C DJNZ R5, rel	2B, 1C MOV R5, A	2B, 1C MOV A, R5
E	1B, 1C INC R6	1B, 1C DEC R6	2B, 1C ADD A, R6	2B, 1C ADDC A, R6	2B, 1C ORL A, R6	2B, 1C ANL A, R6	2B, 1C XRL A, R6	2B, 1C ORL R6, #data	2B, 1C MOV R6, #data	2B, 1C MOV R6, #data	2B, 1C SUBB R6, dir	2B, 1C CJNE R6, #data, rel	2B, 1C XCH A, R6	2B, 1C DJNZ R6, rel	2B, 1C MOV R6, A	2B, 1C MOV A, R6
F	1B, 1C INC R7	1B, 1C DEC R7	2B, 1C ADD A, R7	2B, 1C ADDC A, R7	2B, 1C ORL A, R7	2B, 1C ANL A, R7	2B, 1C XRL A, R7	2B, 1C ORL R7, #data	2B, 1C MOV R7, #data	2B, 1C MOV R7, #data	2B, 1C SUBB R7, dir	2B, 1C CJNE R7, #data, rel	2B, 1C XCH A, R7	2B, 1C DJNZ R7, rel	2B, 1C MOV R7, A	2B, 1C MOV A, R7

Continued...

Appendix B: Special Function Register Format

TMOD : [Bit 0 (LSB) to Bit 3 is for Timer 0 and Bit 4 to Bit 7 (MSB) is for Timer 1]

GATE	C / T	M1	M0	GATE	C / T	M0	M1
------	-------	----	----	------	-------	----	----

GATE: Timer only runs while /INT1 is set.

C / T: '1' for event counter, '0' for interval timer

M1, M0: Mode bit select

"00" Mode 0 – 13-bit timer mode

"01" Mode 1 – 16-bit timer mode

"10" Mode 2 – 8-bit auto-reload mode

"11" Mode 3 – Split timer mode

TCON :

TF1	TR1	TF0	TRO	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

TCON.7 TF1 Timer 1 overflow flag. Set by hardware on overflow.

Clear by hardware when processor vectors to interrupt routine.

TCON.6 TR1 Timer 1 run control bit. Set/cleared by software to start/stop timer.

TCON.5 TF0 Timer 0 overflow flag. Set by hardware on overflow.

Clear by hardware when processor vectors to interrupt routine.

TCON.4 TRO Timer 0 run control bit. Set/cleared by software to start/stop timer.

TCON.3 IE1 Interrupt 1 Edge flag. Set by hardware when interrupt 1 falling edge is detected. Cleared when interrupt is processed.

TCON.2 IT1 Interrupt 1 Type control bit. Set / cleared by software to specify falling edge / low level triggered external interrupts.

TCON.1 IE0 Interrupt 0 Edge flag. Set by hardware when interrupt 1 falling edge is detected. Cleared when interrupt is processed.

TCON.0 IT0 Interrupt 0 Type control bit. Set / cleared by software to specify falling edge / low level triggered external interrupts.

SCON :

SMO	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SMO SM1

0 0 = Shift register mode

0 1 = 8-bit UART mode

1 0 = 9-bit UART mode (Fixed Baud Rate)

1 1 = 9-bit UART mode (Variable Baud Rate)

SM2 = '1' = Enable multiprocessor communication

REN = Receiver Enable

TB8 = Transmit Bit

TI = Transmit Interrupt

RI = Receive Interrupt

Continued...

IE:

EA		ET2	ES	ET1	EX1	ET0	EXO
----	--	-----	----	-----	-----	-----	-----

Bit	Position	Symbol	Bit Address	Description
IE.7		EA	AFH	Global enable/disable. EA = '1', each individual source is enable/disable By setting/clearing its enable bit. EA = '0', disable all interrupts.
IE.6		-	AEH	Undefined
IE.5		-	ADH	Not implemented in 8051. ET2 for 8052.
IE.4		ES	ACH	Serial port interrupt enable bit.
IE.3		ET1	ABH	Timer1 interrupt enable bit.
IE.2		EX1	AAH	External interrupt enable bit.
IE.1		ET0	A9H	Timer0 interrupt enable bit.
IE.0		EXO	A8H	External interrupt enable bit.

IP:

		PT2	PS	PT1	PX1	PT0	PX0
--	--	-----	----	-----	-----	-----	-----

IP.7		-	-					Undefined.
IP.6		-	-					Undefined.
IP.5		-		BDH				Not implemented in 8051. PT2 for 8052.
IP.4		PS		BCH				Serial port interrupt priority bit.
IP.3		PT1		BBH				Timer1 interrupt priority bit.
IP.2		PX1		BAH				External interrupt priority bit.
IP.1		PT0		B9H				Timer-0 interrupt priority bit.
IP.0		PX0		B8H				External interrupt priority bit.

Selected Interrupt Vectors

Interrupt source	Flag	Vector Address
System Reset	RST	0000H
External 0	IE0	0003H
Timer 2 (8052)	TF2 & EXF2	002BH

PSW:

CY	AC	F0	RS1	RS0	OV	-	P
----	----	----	-----	-----	----	---	---

CY : Carry Flag

AC : Auxiliary Carry Flag

RS1, RS0: Register Bank Select

OV : Overflow Flag

P : Parity

End of Paper